REMARKS

Applicants and applicants' attorney express appreciation to the Examiner for the courtesies extended during the recent interview held on October 6, 2003. Reconsideration and allowance for the above-identified application are now respectfully requested.

In the most recent Office Action, dated July 30, 2003, claims 1-25 were rejected either under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,427,161 issued to LiVecchi (hereinafter referred to simply as "LiVecchi") or under 35 U.S.C. § 103(a) as being unpatentable over LiVecchi in view of U.S. Patent No. 6,574,662 issued to Sugiyama et al. (hereinafter referred to simply as "Sugiyama").

By this paper, claims 1, 5-10, 12, 13, 15, 17, 20, 22, 24 and 25 have been amended, and claims 16, 21 and 23 have been cancelled.[1] Accordingly, claims 1-15, 17-20, 22, 24 and 25 are presented for reconsideration, of which claims 1, 17 and 24 are the independent claims at issue.

Method claims 1 and 24 are generally directed to a method (and an associated computer program product, e.g. claim 17, for implementing the method) of a server reducing denials of service even though the server is experiencing a denial of service attack. As recited, the server receives connection requests, establishes a connection socket for one or more of those connection requests without putting that connection request into a backlog queue. For those connection requests for which the server cannot currently establish a connection socket, the connection request is placed in the backlog queue without establishing a connection socket for the time being. The backlog queue includes connection requests without regard for whether or not the connection request includes associated request data. Upon determining that the backlog queue is being used, the server identifies any connections sockets that have no received request data, and disconnects those identified connection sockets.

The pending claims are neither anticipated by nor made obvious by the art of record. In particular, LiVecchi and Sugiyama neither anticipate or nor make obvious the pending claims, either singly or in combination.

In contrast to the pending claims, LiVecchi is directed towards thread scheduling techniques for multithreaded servers (LiVecchi, Title), and is not directed towards countering

[1] It will be appreciated that the claim cancellations made herein should not be construed as Applicants acquiescing to the purported teaching and prior art status of the art of record, namely, LiVecchi and Sugiyama. Accordingly, Applicants reserve the right to further challenge those purported teaching, and/or prior art status of the art of record, at any appropriate time, should the need arise.

denial of service attacks. As part of the thread scheduling, however, LiVecchi teaches various queuing techniques for responding to incoming requests.

For example, with respect to the prior art, LiVecchi states "[f]or applications which receive requests from a number of clients, a special 'passive' socket is created which represents a queue of pending client connections of pending client connections" (Col. 2, lines 62-64). Note that "each client that needs the services of this application requests a connection to this passive socket" (Col. 2, lines 64-66, emphasis added). A "separate 'dispatcher' thread is typically responsible for monitoring the queue which receives the incoming connections requests for the passive socket" (Col. 3, lines 15-18). The server "accepts a pending client connection from the special passive socket" (Col. 3, lines 3-4). This "creates a new server socket, which is then assigned to an available thread for processing" (Col. 3, lines 5-6).

The Office Action appears to treat the passive connection queue of LiVecchi as being the backlog queue recited in the claims. However, when the claim is read as a whole, that is not warranted. For instance, LiVecchi does not teach that a connection socket is established for any of the incoming requests without placing the connection request in the passive connection queue. According to the prior art method described by LiVecchi, each incoming requests is placed in the passive connection queue (see Col. 2, lines 64-66). Furthermore, the prior art method described by LiVecchi does not teach that in response to determining that the passive connection queue is being used, connection sockets are identified that have no received request data, and those identified connection sockets are disconnected.

LiVecchi also describes a "2-stage queue" in which three separate queues are used. Upon receipt of an incoming connection request, the server creates a new socket data structure and an entry for the connection is put in a "pending connections" queue (the first of the three queues) (see Col. 12, lines 13-21). Upon receipt of a confirmation of the connection, the connection is marked as "accepted" and the connection entry is moved from the "pending connections" queue to the "accepted connections queue" (the second of the three queues) (see Col. 12, lines 36-43). Upon the subsequent receipt of a data packet for that connection, the connection entry in moved to the "ready" queue (the third of the three queues) (see Col. 12, lines 43-65). Accordingly, any connection entry in the "ready" queue will already have associated received request data.

The Office Action also appears to treat the "ready" queue of LiVecchi as corresponding to the backlog queue recited in the claims, based on the passages asserted as relevant to the independent claims. However, once again, this is not justified, especially in light of the amendments to the claims. For instance, the "ready" queue only includes those connection entries for which there is associated received request data. It does not include connection requests without regard for whether the connection requests include associated data. Furthermore, LiVecchi does not indicate that when the "ready" queue is being used, the connection sockets that have no received request data are identified, and then disconnected. In addition, LiVecchi does not teach that a connection socket is established for any of the connection request without first placing an entry for the connection request in the "ready" queue.

Sugiyama is directed towards a system for network-device management including collecting a storing of device attributes that change with time and device attributes that hardly change with time (Sugiyama, Title). Sugiyama teaches the use of WinSock, but is otherwise unrelated to responding to connection requests. Accordingly, none of the cited reference, either alone or in combination, anticipate or make obvious the method, and computer program product as claimed.

Accordingly, favorable action is requested. In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 20th day of OCTOBER, 2003.

Respectfully submitted,

RICK D. NYDEGGER
Registration No. 28,651
Attorney for Applicant
Customer No. 022913